August 2020

Content Policy Violation Detection Based On Semantic and Syntactic Image Matching

Anonymous

Content Policy Violation Detection Based On Semantic and Syntactic Image Matching

ABSTRACT

This disclosure describes a two-stage adversarial defense framework for the detection of policy violating content in online content platforms. Deep learned semantic features as well as pixel-level features of an input image are utilized to detect policy violating images. A two-stage template-matching based approach is utilized to detect policy violating images. In a first stage, semantic matching is utilized to search for a set of k-nearest neighbors of an input (new) query image from a previously labeled image database. Policy violating (positive) neighbors are identified from the set of the k-nearest neighbors. In a second stage, instance matching between the query image and its positive neighbor(s) is performed to match local features detected from the query image and the positive neighbor images. Geometric verification of local features of the query image is performed against local features of the positive neighbors. Based on the geometric verification, class labels for the query image are determined and utilized to verify the policy compliance of the query image.

KEYWORDS

- adversarial image
- semantic matching
- instance matching
- syntactic matching
- k nearest neighbors (k-NN)
- geometric verification
- image classification
- local features
- adversarial attack
- content policy
- online advertising

BACKGROUND

Platforms that publish online content, e.g. social media networks, news outlets, game platforms, online marketplaces, etc. institute policies to regulate inappropriate and/or offensive content, e.g., that may be posted by users of such platforms. The policies commonly apply to content, e.g., text and images that appear in online advertisements, news items, etc. uploaded to such platforms.

Manual and/or automated classification is utilized by platforms to exclude/reject policy violating content. However, malicious users often attempt to circumvent the policies by modifying their content, e.g. advertisements, in order to pass through automated policy violation classifiers. Such adversarial attacks are mounted by uploading policy-violating advertisement images that are manipulated/tampered by adding noise, blurring, stripes, scribbles, emojis, memes, etc. Malicious advertisers attempt different manipulations until they successfully place their ad on a platform. Given the volume of content that is uploaded to such platforms, manual verification of policy compliance of content is costly and difficult to scale.

DESCRIPTION

This disclosure describes semantic instance matching (SIM), a two-stage adversarial defense framework for the detection of policy violating content in online content platforms. Per techniques of this disclosure, deep learned semantic features as well as pixel-level features of an input image are utilized to detect policy violating images.

A two-stage template-matching based approach is utilized to detect policy violating images. In a first stage, semantic matching is utilized to search for a set of k-nearest neighbors of an input (new) query image from a previously labeled image database. Policy violating (positive) neighbors are identified from the set of the k-nearest neighbors. In a second stage, instance

matching between the query image and its positive neighbor(s) is performed to match local features detected from the query image and the positive neighbor images.

The input image is labeled, e.g., as a policy violating image or as a policy compliant image, based on overlapping features identified between the local features in the query image and its positive neighbors.

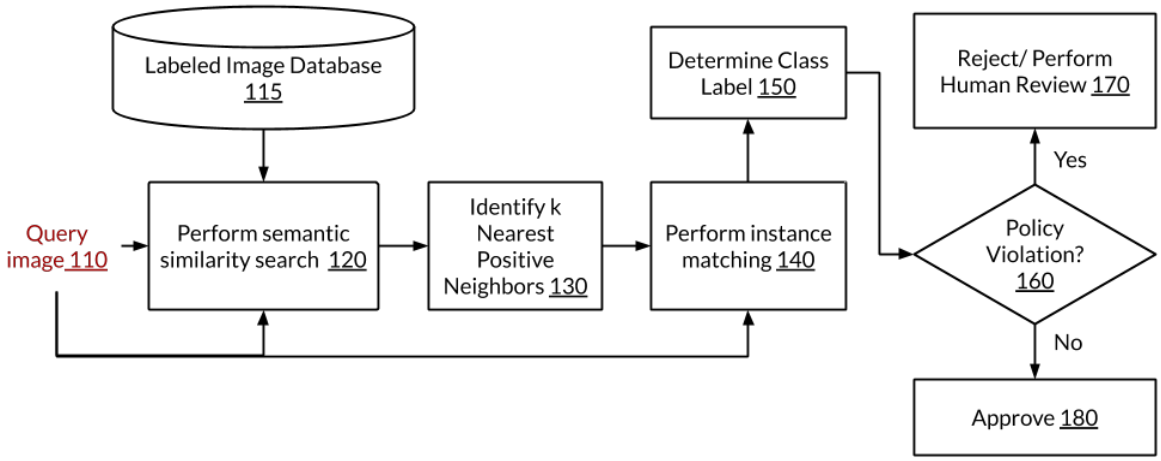

Fig. 1: Policy violating images are detected based on semantic and instance matching

Fig. 1 depicts an example workflow for two-stage detection of policy violating images, per techniques of this disclosure. An input query image is received (110), e.g. based on an ad image uploaded by a user, and is provided as input. A semantic similarity search is performed (120) by utilizing a labeled image database (115) that includes previously classified (labeled) images. The previously classified images are associated with labels indicative of policy compliance/suitability and policy violations, and may be based on previously uploaded images on the platform, or from other sources.

The semantic similarity search is performed by applying a k nearest neighbor (k-NN) search to the query image and retrieving the top k nearest neighbors. Based on the semantic similarity search, a set of k nearest neighbors of the query image is identified. The number of

nearest neighbors identified is a configurable parameter that can be specified, e.g. by an administrator of a content platform.

Based on labels associated with identified k nearest neighbors, a subset of policy violating (positive) neighbors is identified (130). If no positive neighbors are identified by the semantic matching, e.g. the k nearest neighbors to the query image are all suitable/policy compliant, the query image is determined to be policy compliant, and is approved for publishing on the content platform (this path is not shown in Fig. 1).

Next, instance matching is performed (140) between the query image and the identified positive neighbors. An instance-level similarity between the query image and each of the identified positive neighbors is determined. Based on the instance-level similarity, a class label for the query image is determined (150). The newly classified query image can also be added to the labeled image database.

Based on the determined class label for the query image, it is determined (160) whether the query image is a policy violating image. If the query image is determined to be a policy violating image, it is rejected and/or subjected to additional human review (170). If the query image is determined to not be a policy violating image (e.g., based on the class label and/or the human review), the query image is approved (180) for publishing on the platform.

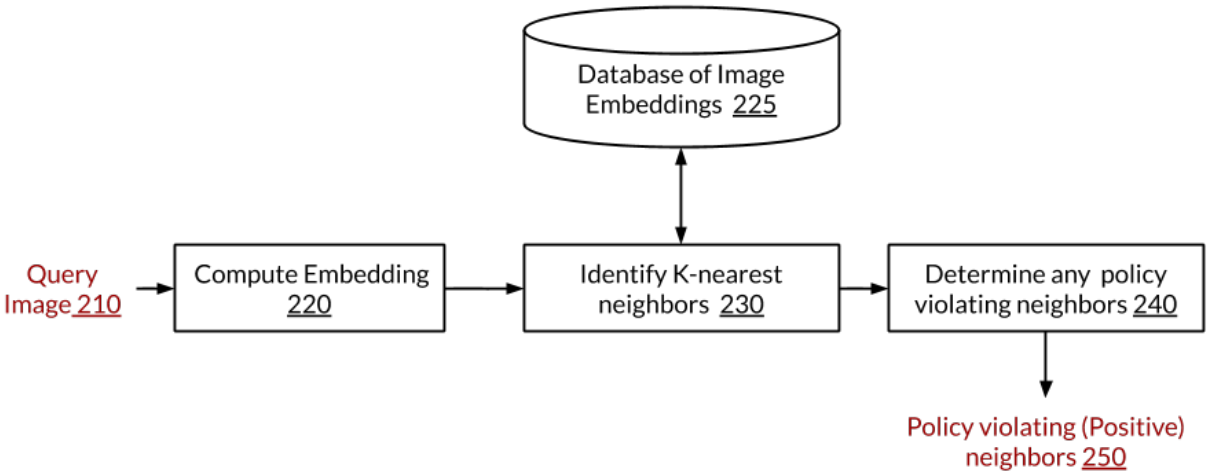

Fig. 2: Semantic matching is utilized to identify policy violating nearest neighbors

Fig. 2 depicts an example workflow for semantic matching of a query image, based on a semantic similarity search, per techniques of this disclosure. The query image is received (210). An embedding, which is a condensed mathematical representation of the query image, is computed for the query image (220). Any type of suitable machine learning model can be used to compute the embedding.

Based on the embedding computed for the query image, its k nearest neighbors are identified (230), based on similarity of the query image embedding with embeddings of images stored in a database of image embeddings (225). In some implementations, global semantic embeddings of the query image and the stored images are utilized as a measure of similarity.

Labels associated with the identified k nearest neighbors are utilized to determine whether any of the identified k nearest neighbors have been previously classified as policy violating (240). Based on the determination, a set of positive nearest neighbors (250), is generated for additional processing, e.g. instance matching, as described above with reference to Fig. 1.

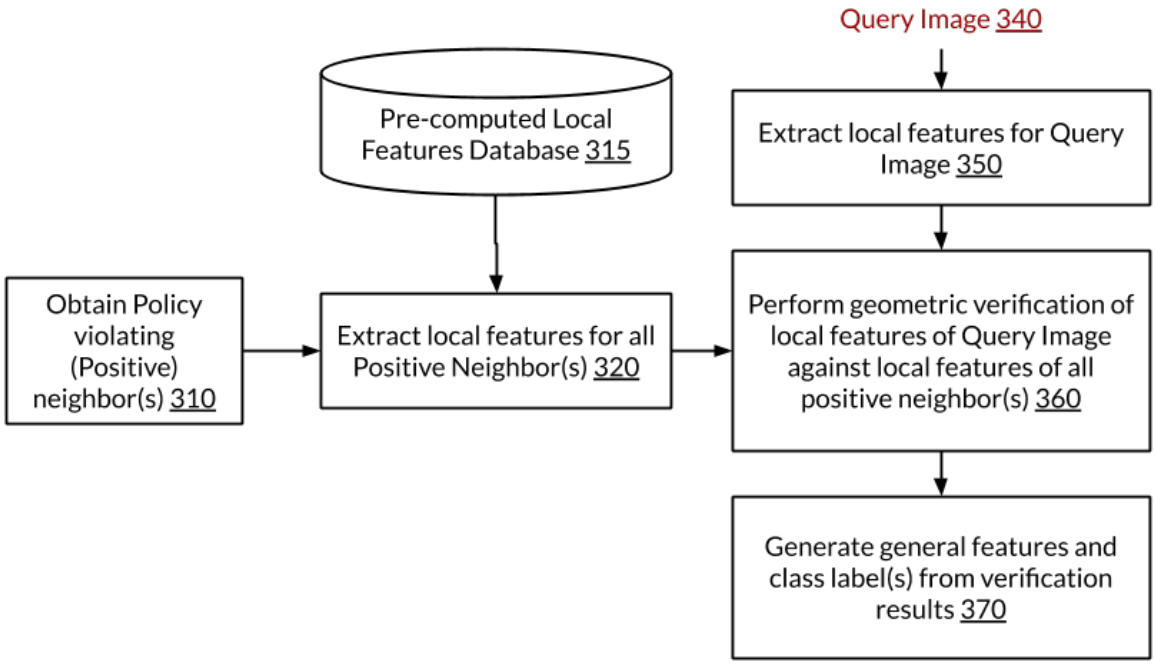

Fig. 3: Instance matching is used to identify overlapping local features

Fig. 3 depicts an example workflow of instance matching to detect policy violating images, per techniques of this disclosure. Instance matching is utilized to identify pixel-level correspondence between any two images and utilizes geometric verification to identify overlapping features between the images.

In this illustrative example, policy violating neighbor(s) images of a query image are obtained (310). The policy violating neighbors can be generated, e.g. based on semantic matching of the query image with its k nearest neighbors, as described with reference to Fig. 2. Local features of the positive neighbor(s) are extracted (320). Local feature extraction may utilize techniques such as Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Oriented FAST and rotated BRIEF (ORB), etc. A database of precomputed local features (315) can be utilized in the extraction of local features.

Local features of the query image (340) are also extracted (350). Geometric verification of local features of the query image is performed (360) against local features of the positive neighbors. Geometric verification is a similarity measure method that operates on local features and returns results indicative of potential overlapping portions between the query image and the positive neighbors. A rigid 2D projective model is utilized to match local features detected from a source image (positive neighbor) to a target image (query image).

In some implementations, techniques such as Random Sample Consensus (RANSAC) may be applied in the geometric verification. Based on the geometric verification, general features and class labels for the query image are determined (370). The determined class labels can be utilized to determine the policy compliance or the policy suitability of the query image, as described earlier.

The two-stage framework that includes semantic and instance matching stages, per techniques of this disclosure is a robust technique for the detection of policy violating images, particularly of adversarial images, and can handle the various image modifications that an adversary may attempt to circumvent automated checks for policy violations.

Semantic matching, e.g. as depicted in Fig. 2, enables reduction in the number of false positives, e.g. images flagged by an algorithm as policy violating but that are not actually policy violating. For example, some textures like watermarks are likely to cause false positives in prior techniques, as they can appear in both suitable and policy-violating images.

Additionally, the use of semantic matching can be performed with a lower computational workload than prior techniques. Instead of applying instance matching to all positive images in the database, semantic matching is utilized to construct a limit-sized search vicinity for the query image.

Instance matching, as described with reference to Fig. 3, enables detection of policy violation even if it is due to a small portion of the input image. For example, a carefully designed malicious ad image could have a suitable (policy compliant) background and a small foreground object which violates the policy. Such an image could be difficult to detect based just on semantic matching. However, instance matching can detect such a violation, provided that local features of the (violating) foreground object are included in the database of policy violating images. Further, violating images can be added to the database, improving the robustness of the described techniques over time.

Local features are robust to translation, rotation, scale, lighting, pose, and viewpoint changes, and thus enable instance matching to detect adversarially manipulated images. Instance matching is thus robust to adversarial attacks that rely on modification of such local features. Adversarial attacks like blurring or adding stripes that can pose a challenge to semantic matching techniques, e.g. by causing computation of global image embeddings that are distant from the original, unmanipulated images, can still be accurately detected by instance matching.

Instance matching also enables good detection performance even in the presence of noisy (erroneous) labels. Erroneously labeled images in the immediate semantic neighborhood of a query image can pose a challenge to semantic matching. The use of instance matching enables accurate detection, as long as at least one correctly labeled neighbor is found in the k-NN space.

Techniques of this disclosure can be utilized to detect adversarial policy violating content, e.g. advertisements that are commonly missed by conventional, e.g. single-stage classifiers, or other techniques. The two-stage framework described herein can be customized by replacing default global image embedding and local features with other embeddings or features, e.g., deep-learned local features.

The described techniques can be used in combination with other factors for identifying or handling content policy violations, e.g., other image processing techniques to analyze the query image, user information for users associated with uploaded images, etc.

CONCLUSION

This disclosure describes a two-stage adversarial defense framework for the detection of policy violating content in online content platforms. Deep learned semantic features as well as pixel-level features of an input image are utilized to detect policy violating images. A two-stage template-matching based approach is utilized to detect policy violating images. In a first stage, semantic matching is utilized to search for a set of k-nearest neighbors of an input (new) query image from a previously labeled image database. Policy violating (positive) neighbors are identified from the set of the k-nearest neighbors. In a second stage, instance matching between the query image and its positive neighbor(s) is performed to match local features detected from the query image and the positive neighbor images. Geometric verification of local features of the query image is performed against local features of the positive neighbors. Based on the geometric verification, class labels for the query image are determined and utilized to verify the policy compliance of the query image.

REFERENCES

1. Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60, no. 2 (2004): 91-110.
2. Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski. "ORB: An efficient alternative to SIFT or SURF." In *2011 International conference on computer vision,* pp. 2564-2571. Ieee, 2011.
3. Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." In *European conference on computer vision,* pp. 404-417. Springer, Berlin, Heidelberg, 2006.
4. Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." *Communications of the ACM* 24, no. 6 (1981): 381-395.